

## APPLICATION OF COMPUTATIONAL PHYSICS WITHIN NORTHROP

M. W. George, R. T. Ling, J. F. Mangus  
Northrop Corporation, Aircraft Division  
Hawthorne, California

W. T. Thompkins  
Northrop Research and Technology Center  
Palos Verdes, California

An overview of Northrop programs in computational physics is presented. These programs depend on access to today's supercomputers, such as the Numerical Aerodynamic Simulator (NAS), and future growth will depend on the continuing evolution of computational engines. Descriptions here are concentrated on the following areas: 1) computational fluid dynamics (CFD), 2) computational electromagnetics (CEM), 3) computer architectures, and 4) expert systems. Current efforts and future directions in these areas are presented. The impact of advances in the CFD area is described, and parallels are drawn to analogous developments in CEM. The relationship between further advances in these areas and the development of advanced (parallel) architectures and expert systems is also presented.

## BACKGROUND

The design of modern fighter aircraft requires the analysis of many factors. Some of these, including complex flow phenomena and electromagnetic characteristics, pose serious problems for the designer. The analysis of these conditions is complicated by the fact that it is extremely expensive and time consuming to reproduce them in an experimental test environment. Fortunately, computational methods have advanced sufficiently to allow mathematical simulation of these phenomena, supplementing physical testing during the design process.

The trend towards integrating computational methods into the design process has been driven by the rapid advances made in computer hardware over the last decade. This trend has resulted in the development and application of methods capable of analyzing complete aircraft configurations.

Over the past ten years at Northrop, computational physics had its largest impact in the area of computational fluid dynamics. This began with a requirement to analyze transonic flow phenomena for fighter aircraft. In 1975 Northrop applied the 3-D Bailey-Ballhaus transonic small disturbance code developed at NASA Ames (Ballhaus, Bailey, and Frick [1976]) to the solution of this problem. This was followed by the application of the full potential codes, leading to our present use of Euler and Navier-Stokes codes. Throughout this period of development, Northrop efforts were enhanced by cooperative programs with NASA. These programs gave Northrop access to state-of-the-art computer facilities that allowed for the advanced development of methodology and application techniques that would be assimilated into Northrop's evolving project areas.

Our initial runs with the transonic small disturbance codes were done on an IBM 360 series computer. These facilities could barely handle a wing-alone configuration. On today's computers, those early codes can be run numerous times a day. Today, the Euler and Navier-Stokes codes are straining the capabilities of our equipment, which is limiting the degree to which these methods can be beneficially applied. As in the past, Northrop is working with NASA through cooperative programs to develop applications of these methods on leading-edge computing facilities.

Throughout the last decade several factors have stimulated Northrop's development and application of computational methods. These have included:

- The insight into the physics of the flow that was obtainable from computational methods

Prior to extensive use of computational methods, the design of aircraft was based primarily on experimental testing, in particular, wind tunnel testing. Wind tunnel testing typically generates global values (lift, drag, moments) with limited surface pressure measurements, flow visualization and flowfield measurements. It is very expensive in both time and money to acquire more extensive flowfield information, such as off-body information, through experimental testing. Computational methods provided a way to see the whole picture, yielding an excellent complement to experimental testing. With this ability to see the flow phenomena in total came a better understanding of the physics of problems, providing a more complete and valuable understanding to the design process.

- The emerging emphasis on low observables

The shifting emphasis to low observable characteristics forced new aircraft configurations towards shapes and concepts outside of the established data base. To do a parametric study on the new families of geometries, using the traditional approach of experimental testing, was prohibitive in both time and money. Computational fluid dynamics filled this void in the aerodynamic analysis of new radar cross section (RCS) driven configurations. As a result, they earned acceptance in the design process.

- Testing limitations

Wind tunnel testing cannot accurately simulate flight conditions, especially in the transonic flow regime, where wall effects, mount effects, and scaling have a significant impact on flow characteris-

tics. The ability of the codes to model both free-flight conditions and test conditions results in a powerful tool for better understanding test results, and for extrapolating results to free-flight conditions.

In brief, Northrop's acceptance of these codes was the result of their ability to: 1) increase our understanding of flow phenomena associated with conceptual designs, 2) expand the number of design parameters while controlling costs and 3) improve the effectiveness of test programs.

### Northrop Applications

The emphasis within Northrop, in the area of computational fluid dynamics, has shifted almost entirely to the application of the Euler and Navier-Stokes equations. The flight environment of fighter aircraft, which is dominated by a multi-sonic vortex/viscous environment, requires a level of physical detail that can only be supplied by the Euler and Navier-Stokes methods.

As a result of the advances made in the area of computational fluid dynamics in the last decade, an emerging area for the application of computational physics at Northrop has been in the solution of the Maxwell/Helmholtz equations. The similarity of the Maxwell equations to the Navier-Stokes equations allows the advances made in computational fluid dynamics to be applied to the solution of electromagnetics problems. Advances in algorithms, grid generation techniques, and specialized computer architectures, developed for computational fluid dynamics, are all readily applicable to the solution of the Maxwell equations.

### Future Directions

The application of both computational Navier-Stokes and Maxwell methods to full configurations in the actual design environment is now a realistic goal. Achieving that goal will require, in addition to further algorithm development: (1) the further development of computer hardware and architectures, and (2) the development and application of expert systems to make productive and efficient use of this potential.

Useful solutions of the Reynolds-averaged Navier-Stokes equations for flow about a realistic aircraft configuration would require hours of run time on one of the current generation of supercomputers. Because the aircraft designer must study a large number of configurations, even an hour's wait for a solution is unacceptable. In addition, the high cost of such computers prohibits running them in this manner. As a possible solution to this resource problem, Northrop is studying the application of parallel architectures. By solving a large, computationally-intensive problem on a system with multiple processors working concurrently, the solution time can be reduced by a factor approaching the number of processors. In addition, parallel processing provides an advantage of flexibility in the allocation of computer resources. As small projects develop, computational resources can be increased by adding dedicated processors. While parallel processing architecture promises a viable approach to the computer resource problem, it requires a rethinking of solution algorithms.

The potential value of current and evolving computational physics codes, coupled with the development of new computers, is immense. The resulting systems will be able to analyze multiple families of parameters, exploring new and innovative concepts in relatively short time periods. The data generated in this process will be overwhelming. In order to effectively utilize these systems,

the integration of expert systems will be required. Northrop is currently addressing the need for expert design systems as part of its program in computational physics.

While Northrop is developing and applying computational physics methods to other areas (e.g. structures, avionics, controls, simulation, etc.), fluid dynamics and electromagnetics place the largest demands on computational resources. This paper will therefore address Northrop's current programs in these two areas, along with corresponding efforts in the area of computer architectures, and expert systems.

## COMPUTATIONAL FLUID DYNAMICS (CFD)

Over the past decade, the aerodynamic and propulsion design processes have undergone a significant change. Where the process was once dominated by the use of wind tunnel facilities as analysis and design tools, the trend today is to rely more on CFD methods as the principal design tools. There are several reasons for this change. Wind tunnel testing has always been expensive, in terms of manpower, time, and facilities. Over the last ten years these costs have increased. At the same time, the development of sophisticated CFD methods and computing engines have made these methods more efficient and effective. In comparing the results gained by the two approaches, computational techniques are also gaining an advantage. The inherent limitations of wind tunnel testing, including the restrictions imposed by modelling, wall effects, etc., do not constrain today's CFD methods. Instead their application has been limited by the available computer hardware. With advances in "supercomputers", those limitations are rapidly being overcome.

Northrop's experience with CFD methods has been strongly influenced by the specific complexities of fighter aircraft design. Fighter aircraft incorporate geometrical features such as closely-coupled lifting surfaces, sharp or small leading edge radius, and vortex generating devices as part of a multi-point, multi-sonic design emphasizing high angle-of-attack maneuverability. These features generate a flow environment strongly dominated by viscous and vortex phenomena. Also, the strongly coupled flow environment limits the utility of the application of CFD methods to individual components. In this environment, earlier methods such as transonic small disturbance and full potential could be applied to fighter design primarily in regimes where the flow "behaved nicely", such as under cruise conditions. Application of computational methods to full fighter configurations over the complete design envelope required both more sophisticated methods, and more capable computing resources.

As a result of Northrop's emphasis on fighter design, CFD efforts have been directed towards Euler and Navier-Stokes methodology. Today, computational Euler methods are utilized on a daily basis in design projects for application to configurations as well as to isolated components. The computer resources within Northrop which provide the capability to run the Euler methods are two FPS-164 (Floating Point Systems) computers. These machines have in-core memory of four and seven million 64-bit words, which allows for the modeling of meaningful configurations. These machines have enabled the development of Euler methods at Northrop and their integration into the design environment. They have also allowed for initial exploration of Navier-Stokes methods.

The current capabilities and limitations of Euler methods are illustrated in figures 1 to 7. As stated previously, the primary

reason for the selection of Euler methodology is its ability to model the vortex-dominated environment associated with fighter aircraft, as indicated in figures 1 to 4. The Euler methods that generated the following results are based on the finite volume formulation developed by Jameson, et al (1981).

Figure 1 illustrates configuration and flow capabilities provided by current Euler methodology. The configuration in Figure 1a was generated by combining a chined forebody (Erickson and Brandon [1985]) together with the wing from the AFVFM (Air Force Vortex Flap Model, Erickson [1985]) and a tail representative of the F-18. A configuration similar to this will be tested in 1987 as part of a cooperative program between Northrop and NASA Ames. Figures 1b and 1c show the complexity of flow associated with this type of configuration, specifically the vortex-dominated environment, composed of interacting vortices generated by the chine, wing and tail. The total pressure contours in Figures 1b and 1c show the vortex structure at a wing-body fuselage station and tail-body fuselage station, respectively.

Figures 2 and 4 show the capabilities of the Euler methods to calculate total forces on fighter-type configurations. Lift versus angle-of-attack results for both Euler and experiment are compared in figure 2 for the AFVFM shown in figure 3. The agreement in both value and location of  $C_{LMAX}$  is quite good. As is typical of this type of fighter configuration, the stall characteristics are due to the burst point of the vortex generated by the wing leading edge moving forward over the wing. The nonlinear lift effects which are also associated with the wing leading edge vortex passing over the wing, and which contribute to the increasing lift curve slope prior to stall, are also modeled by the Euler code. The experimental results showed wing tip separation occurring near 15 degree angle-of-attack, which accounts for the early decrease in lift curve slope as compared to the Euler results.

Figure 4 shows comparison of experimental and Euler drag polars for the F-20 Tigershark at a Mach number of 0.80. As seen in figure 4, the comparison between experiment and computation is quite good over the entire range. Figure 4 was taken from the paper by Bush, Jager, and Bergman (1986), which gives more comprehensive coverage of Euler code application within Northrop.

Another area in which Euler codes are used extensively is in the design and analysis of inlets. The ability of the Euler methods to correctly model shock structure and corresponding total pressure losses makes them applicable to the transonic and supersonic inlet problems. Figure 5 shows the results of an Euler method applied to a 3-D supersonic compression-ramp inlet. The ability of the Euler method to model the shock structure is shown in this figure. The pressure contours show the shock emanating from the ramp combining with the standoff shock from the inlet cowl.

While the current capability of the Euler codes is proving to be extremely useful in the design environment, their limitations due to lack of viscous modeling and computer resources is placing an increased demand on the development/acquisition of larger computing engines and on the development/application of Navier-Stokes methods. While the creation of the vortex structure and its resultant gross effects on the aerodynamic characteristics are not that sensitive to mesh size, the local effects (shock-vortex interaction, surface pressures, vortex empennage interactions, etc.) are not accurately modeled in the Euler methods without adequate grid definition. The addition of viscous terms is needed to improve the calculation of drag, total pressure losses (inlets), entrainment (nozzles), and separation phenomena. Figures 6 and 7 show some of the deficiencies due to current computer resources and lack of viscous modeling.

Figure 6 shows the comparison of experimental and Euler pressure results at a forebody station of the chined forebody configuration in figure 1a. As seen in this figure the Euler results do not resolve the peak pressure due to the vortex (formed from the chine) which sits over the body. The discrepancy is due to the lack of grid resolution in defining the vortex, as shown by Rizzi (1985).

The comparison of experimentally obtained pressures and pressures calculated by the Euler code are shown in figure 7 for a spanwise cut on the AFVFM, with the leading edge flap deflected 30 degrees. Experimental results are shown for both 0.4 and 0.7 Mach numbers. The primary vortex generated by the wing leading edge and the secondary vortex structure generated at the flap hinge line are not modeled well by the Euler code due to a lack of grid resolution and lack of viscous modeling.

Northrop's current computer resources do not yet allow the application of Euler and Navier-Stokes codes to complete fighter aircraft configurations. Another limitation is the current inability to grid the complete configurations. The capability of Euler and Navier-Stokes methods to adequately model a full configuration depends on a grid scheme that provides an arrangement of points to discretize the equations and model the physics. For simple configurations this process is easily accomplished, but for problems such as complete fighter configurations this becomes the most difficult part of the solution procedure. The AFVFM (figure 3) and the chined wing-body configuration (figure 1) were gridded as illustrated in figure 8. The grid is generated by defining a series of "C" type surface grids which extends forward and rearward of the wing. The portions of the surface grid forward and rearward of the wing can accommodate chines, leading edge extensions (LEXs), tails and flow-through conditions. The boundary grids and internal grids are generated by a combination of 2-D and 3-D Poisson and transfinite interpolation solvers. This grid approach, due to its contiguous nature, is limited in its applications. The current trend is to develop grid generation techniques which utilize a block structure and grid lines that are not necessarily continuous within or across blocks.

Northrop's current and future activities in the area of Navier-Stokes and Euler development involve cooperative efforts with NASA. As stated in the introduction, these cooperative programs allow access to the evolving "supercomputers" along with access to NASA personnel and methods. A current NAS program being done in conjunction with the Ames Research Center involves the application of the TNS code (Kaynak, Holst, and Cantwell [1986]) to the AFVFM. The AFVFM provides a good test basis for determining the applicability of Navier-Stokes and Euler methods to fighter configurations. The AFVFM (figure 3) provides a simple, easily gridded geometry which generates some of the primary vortex flow phenomena associated with fighter configurations. The AFVFM incorporates a swept wing with a series of leading edge vortex flaps (including sharp and round leading edges) and conventional trailing edge flaps. In addition to this NAS program, cooperative efforts exist in the areas of the development/application of Navier-Stokes methods to nozzles and the application of Euler methods to chined forebody configurations.

The growing importance of and demand for CFD in the design and project areas within Northrop is placing more emphasis on the development of Euler and Navier-Stokes methods and acquisition of more capable computing systems, along with the development of pre- and post-processing techniques. We expect that, with the current growth in computer technology and the evolving methodology, Navier-Stokes methods (Reynolds averaged) and the systems to utilize them will be commonplace within Northrop in the next five years.

## COMPUTATIONAL ELECTROMAGNETICS (CEM)

Requirements for aerodynamic performance and electromagnetic characteristics, such as radar cross section (RCS), have become critical drivers in the design of modern military aircraft. Northrop has been actively involved in RCS analysis for more than twenty years. It has been a pioneer in the development of Physical Theory of Diffraction (PTD), and integral equation methods. The MISCAT/GENSCAT codes, developed by Northrop under a series of contracts to government agencies, have found widespread usage by many aircraft and missile manufacturers. These codes, like the panel method codes in CFD, are forerunners of the emerging field of computational electromagnetics (CEM). The latter is as important in the design process as computational fluid dynamics (CFD), figure 9. Just as CFD codes can be considered as numerical wind tunnels, CEM codes can be considered as numerical radar ranges and anechoic chambers. As CFD plays an increasingly important role in supplementing costly wind tunnel testing in the design process, so CEM is expected to supplement expensive testing in radar ranges and anechoic chambers.

Actually, the similarity between CFD and CEM runs much deeper than this. Since both the aerodynamic performance and the electromagnetic characteristics are configuration dependent, a unified approach can be devised for both aerodynamics and electromagnetics problems, figure 10. Starting with the same aircraft configuration, common geometry definition and grid generation procedures can be used in preparation for the solution of respective governing equations to obtain the aerodynamic and RCS characteristics. The most interesting and important aspect of the unified aero/RCS approach lies in the mathematical similarity between aerodynamics and electromagnetics problems, figure 11.

The scattering of electromagnetic waves by an aircraft can be formulated as a boundary value problem analogous to the fluid dynamic problem of flow past the aircraft. The governing Maxwell/Helmholtz equations can be solved numerically in a manner similar to the solution of governing fluid flow equations such as the Navier-Stokes/Euler equations. The far field radiation condition and media interface boundary condition can also be enforced in a manner similar to the enforcement of the freestream condition and flow tangency condition. Various numerical methods in CFD can be carried over for the computation of electromagnetic characteristics.

For the electromagnetics problem, the Maxwell/Helmholtz equations can be solved by differential equation and integral equation methods. One type of differential equation method (King, et al [1959] and Bowman, et al [1969]) involves separation of variables in specific coordinate systems. In the past few years, there have been attempts (Bayliss, et al [1982], [1982] and [1983]) to directly solve the Helmholtz equation for scalar scattering problems by finite difference methods. Besides these finite-difference, frequency-domain (FD-FD) methods, a finite-difference, time-domain (FD-TD) method has also been proposed (Umashankar, et al [1982]).

In the integral equation approach to scattering problems, an equivalent integral equation such as the Chu-Stratton equation containing Green's function can be derived for either the magnetic field or electric field and can be formulated either in the time domain or frequency domain (Mitra [1974]). Many techniques are available for reducing the integral equation to a matrix equation for numerical solution. Some of these are grouped under the

title of "moment method" (Harrington [1968]). Moment method codes in general are not numerically efficient. At Northrop, they are mainly used to validate results obtained from new methods under development, for obstacles of simple geometry.

Numerical techniques developed for CFD have various degrees of applicability to nearly every CEM method mentioned above. However, the CEM method that is most closely related to common CFD methods – and therefore best suited for exploiting the advances made in CFD – is the finite difference, frequency domain (FD-FD) method based on the concept of generalized scattering amplitude (Ling [1986] and [1987]). In this CFD approach to solving electromagnetics problems, the original Helmholtz equations in terms of electric and magnetic field vectors are transformed into scalar equations in terms of generalized scattering amplitudes or related Debye amplitude functions.

The current CEM research at Northrop consists of development of methods in both the integral equation and differential equation approaches. In the former, efforts are focused on the k-space method (Bojarski [1971]). The aim here is to find efficient iterative procedures to make it practical in the design process. A CFD approach developed at Northrop has recently been applied to simple obstacle shapes including the circular cylinder and sphere (Ling [1986] and [1987]), figures 12 and 13. Numerical results agree with the exact eigenfunction expansion solutions (King [1959] and Bowman, et al [1969]).

To apply CFD methods to practical RCS problems involving complex aircraft geometry and incidence of high frequency electromagnetic waves, fast processing capability and large memory storage of a supercomputer such as the NAS are required. This is due to the large number of grid points necessary for resolution of the scattering characteristics generated by complex geometry aircraft. Though the introduction of the radially non-oscillatory generalized scattering amplitude has largely eliminated the need to resolve the field quantity oscillations along the radial distance to infinity, one still has to contend with the generalized scattering amplitude variation inside the finite volume of the aircraft. To achieve sufficiently accurate resolution of field quantities and generalized scattering amplitudes, the grid spacing should be one-tenth of a wavelength or less. Representative RCS problems encountered in the design process may involve obstacles with dimensions on the order of one hundred wavelengths or more in each of the three directions. This would require dealing with millions of grid points. Solution of matrix equations for such a large system certainly requires a supercomputer.

The CFD approach to electromagnetic wave scattering shows promise for accurate, systematic, and efficient calculations for obstacles of arbitrary material properties, size and shape. The power and capability of a supercomputer can transform CFD methods into practical tools for solving the RCS problems encountered in the military aircraft design process.

## COMPUTER ARCHITECTURES

Northrop's activity in the area of computer architectures is focused on the hardware used in parallel processing. Parallel computing uses multiple processors to simultaneously execute individual parts of a larger overall task. Although this adds to the complexity of the application program, it can greatly decrease its execution time. Within the scope of Northrop's program, parallel processing is viewed as a possible alternative to supercomputers for the small project environment, and also as a concept applicable to supercomputing and other emerging computer technology.

One reason for investigating parallel architectures stems from the unique computing requirements of the Northrop engineering environment. The majority of Northrop's projects operate their own isolated computing facilities. Individual projects cannot provide the computer resources for running codes that require the power of a supercomputer. Purchase of individual supercomputers for many such projects is clearly not cost effective, and security requirements prevent the use of supercomputing resources available at NASA and other government research centers. In this environment, parallel computers offering significant power at reduced cost will enable projects with limited resources to utilize advanced computational methods.

A second reason for exploring concurrent architectures is the machine speedup they offer. As explained by Denning (1985), the maximum speed of a single processor is limited to approximately 1 GFLOPS by the speed of light. Running complex computational physics applications quickly enough to be useful in the design process will require computational speeds in excess of this figure. The only way to attain such speeds will be through the use of some form of parallelism. This trend is already apparent in supercomputers such as the CRAY-2 which uses 4 processors, and is expected to grow in the future. As concurrent processing becomes more prevalent, the knowledge now being developed in this area should enable Northrop to continue making effective use of evolving computer technology.

The Northrop effort examines parallel processing architectures with the ultimate goal of implementing computational physics codes on such machines. Our objectives in this area are as follows: 1) learn how best to parallelize computational physics algorithms, determine those algorithms most suited to concurrent execution, and examine how parallelization affects algorithm behavior; 2) examine what special demands are placed on pre- and post-processing facilities (e.g. grid generation and analysis of results) by parallelized codes; and 3) explore the effects of different computer architectures on the parallelism of specific algorithms and, conversely, what specific architectural details enable optimal performance of concurrent computational physics applications.

Of these goals, the last has proven to be the most complicated and extensive in scope. Architectural issues such as memory organization, interprocessor communication speed and configuration, and processor power will have a large impact on an algorithm's parallel execution. The most important architectural concern has been the degree of coupling between multiple processors and memories. At one end of the range of architectural options, a parallel computer can have a number of processors addressing a single global memory in a tightly coupled system. As the number of processors in the system increases however, the time overhead incurred by many processors accessing one memory over a limited bandwidth data channel will degrade any potential parallel speedup. Consequently, existing shared memory machines seem to be limited to 2-8 processors.

Computers using more than this number of processors employ memories divided into small parts, each accessible by a single processor. This creates a loosely coupled system with limited memory access, and leads to a question of which processors should communicate with which others. The "crossbar" interconnect, in which each processor node communicates with every other processor in the system, becomes prohibitively complex and expensive as the number of nodes grows, eventually giving way to less extensive communication schemes. These range in complexity from simple nearest-neighbor schemes to the binary hypercube structure in which each of  $2^n$  processors is connected to  $n$  other processors, forming one corner of an  $n$ -dimensional cube. The degree to which processors are coupled can be further decreased

until one reaches the other extreme of an uncoupled system in which individual nodes no longer communicate. These simplified interconnection schemes make loosely coupled architectures more desirable for massively parallel systems incorporating hundreds and possibly thousands of processors.

The choice of an efficient interconnection scheme depends largely on the desired application. For example, it is questionable whether a fully implicit finite difference algorithm which parallelizes efficiently on a shared memory machine will do so on loosely coupled architectures. In addition, the resolution of other architectural issues such as the bandwidth of the interconnecting communications channels, the power of individual nodes and the size of individual processor memory is also highly application dependent. In essence, one would like to balance all of these factors to create an optimal configuration for the efficient execution of a given algorithm. Ideally, a variety of applications might then be found to share some roughly similar optimal architecture which would then define our needs for a parallel processor. Whether this is a realistic goal remains to be seen.

The investigation of these issues has proceeded along two avenues, the first being the implementation of a target code on several commercially available parallel computers. This effort was undertaken to gain experience in working with a variety of parallel architectures, with emphasis on examining the relationship between architecture and parallel implementation as well as relative advantages and disadvantages between alternative architectures. It was also deemed useful to gain experience with the different ways of coding parallelism while also measuring the performance improvements gained by these various methods.

To accomplish this, two- or three-dimensional versions of an explicit Euler code which uses the finite volume algorithm due to Jameson, et al (1981) have been ported onto several parallel machines. This code is a particularly good test case because, while primarily explicit in nature, it also incorporates an implicit residual smoothing scheme, enabling us to also examine the parallelism constraints associated with implicitness.

The code was first implemented on the CRAY X-MP/48 supercomputer which uses the shared-memory, tightly coupled architecture described previously (the CRAY architecture is discussed in more detail by Hwang [1985]). This activity was part of an ongoing cooperative program in computational methods between Northrop and Cray Research, Incorporated. The code was converted for parallel execution with minimal modifications using the Microtasking facility available on Cray FORTRAN. After modification, the code executed with a speedup approaching the maximum predicted by Amdahl's law. This demonstrated that fluid dynamics algorithms of this kind may be efficiently implemented on global memory, tightly coupled machines with relative ease. In addition, running on the CRAY machine afforded an opportunity to perform CFD calculations involving very large numbers of grid points, which in turn pointed out some of the specialized post-processing capabilities needed to cope with the resulting large volume of flowfield data. It was found that high resolution color graphics such as the example reproduced (in black and white) as figure 14 (generated with the assistance of personnel from the Applications Department of Cray Research Incorporated) were most valuable in interpreting the resultant data. The variations in shading in this figure (different colors in the original) correspond to the magnitude of the crossflow velocity component near the wing-body-chine model's surface and clearly show wing and chine vortex formation. Advanced graphics capabilities of this kind are a requirement for efficient use of the sorts of computational devices under discussion here.

## EXPERT SYSTEMS FOR DESIGN APPLICATIONS

Additional efforts at code implementation are currently being directed toward two parallel devices, the Butterfly machine produced by Bolt, Beranek and Newman, and the T-20 computer built by Floating Point Systems. The Butterfly (described in detail by Schneck, et al [1985]) is a 16 processor, local memory machine which represents something of a hybrid architecture by virtue of its unique interconnect scheme. Because each processor is connected to a butterfly switching network (shown in figure 15) which enables it to communicate with any one other processor at a time, any processor has potential access to any other local memory, thus making the memory appear global. A 2-dimensional Euler algorithm has been implemented on this machine and has shown a speedup in execution which scales linearly with the number of processors used. In addition, the 3-dimensional version of the algorithm is being implemented and will be used to investigate optimal coding and memory allocation strategies for this architecture. The T-20 computer (detailed by Frenkel [1986]) consists of 16 processors linked in a hypercube structure as shown in figure 16. One of the unique aspects of this machine is its use of the OCCAM programming language, which is specialized for parallel processing. Current work concentrates on implementing the 2-dimensional Euler algorithm on this machine in OCCAM, and has validated the loosely coupled hypercube architecture as appropriate for this type of application.

The second approach to the architecture study has been through participation in the design of a parallel processor system. Under a contract with DARPA, Northrop is working with Paragon Pacific, Incorporated to build a parallel computer specialized for the solution of computational physics problems. Called the Custom Architected Parallel Processing System (CAPPS), the machine will incorporate several innovative features, the most important of which is a user-configurable interconnect structure. This flexibility will increase the utility of this machine for solution of a variety of problems in computational physics.

To aid in the design of this device, the 3-dimensional Euler code described previously has been partitioned to execute in a concurrent blocked-grid mode and is being used to define interprocessor data transfer requirements. Since the algorithm operates in parallel on several subdomains of the main grid, data must be transferred at the subgrid interfaces at selected times. To reduce the time penalty associated with this communication, the frequency with which this transfer takes place has been progressively reduced, and the results on convergence rate observed. As shown in figure 17, overall convergence remains largely unaffected even when communications are reduced by a factor of 10 over the sequential case. This information can now be used to help formulate the requirements for interconnect bandwidths and individual processor speed for the CAPPS and the optimal level of parallelism to incorporate in this algorithm.

In summary, the Northrop program in parallel architectures has had the overall effect of increasing our understanding of the hardware of parallel processing as well as the basic method for introducing concurrency into computational physics algorithms. Specifically, the results show that a loosely coupled architecture such as the hypercube is an acceptable choice for explicit fluid dynamics algorithms. In the future, as additional parallel machines are examined and new parallel algorithms are implemented, the conclusions drawn should provide an accurate picture of how best to incorporate parallel processing into the aircraft design environment.

In the past, aircraft design has generally suffered from a lack of analytical or computational predicative capability. That is to say our capacity to conceptually design aircraft has exceeded our capacity to predict performance without extensive wind tunnel testing or full-scale prototypes. Modern computational methods, when coupled with continued advances in computer power, have the potential to rectify this situation. That is to say our capacity to compute performance and perform mission simulation may exceed our intuitive design ability.

While this viewpoint may strike some as optimistic, it is already true in some important areas.

- fluid dynamics

intuitive understanding of vortex augmented lift vs. detailed description of wing pressure distribution

- structural analysis

intuitive understanding of the structural response of uniform, metallic parts vs. detailed description of stress levels in complex three dimensional shapes

- low observable vehicles

intuitive understanding of low observable trade-offs vs. integration of aerodynamic, electromagnetic, structural, and materials design techniques

- mission analysis

intuitive understanding of factors affecting fighter maneuverability vs. integration of aerodynamic performance, observable, avionics, cost, and maintainability into mission effective vehicles

Since the required performance of future vehicles demands that the benefits identified by increased computational capabilities actually be achieved, we are moving rapidly into system complexity which can only be achieved through close interaction of design engineers with extensive computational facilities.

While our capability to numerically analyze complex or innovative configurations has grown rapidly, a commensurate increase in our capability to integrate this in the design process has not occurred. Increases in aircraft performance have simply not kept up with increases in available computer power.

The difficulty is not a lack of knowledge in the disciplines but rather a lack of capability to encapsulate knowledge and share it among personnel with diverse backgrounds. While the techniques of expert systems and their promise has received far more attention than their current merit would justify, accomplishing effective knowledge sharing, to which the best of these systems are directed, is exactly the task which must be accomplished in order to efficiently cope with the diverse requirements of modern aircraft design.

Thus, the challenge is to develop software systems or environments which can aid designers in their complex technology integration and design tasks.

## Design Process Review

A simplistic view of a top-down aircraft design cycle takes the following form: The process begins with system specifications, and first derives: 1) a functional decomposition among subsystems, 2) subsystem performance goals, and 3) a preliminary system design. Next, a sequence of increasingly detailed iterations between aerodynamic, observable, and structural requirements is conducted to define a detailed geometric shape. Iteration between performance goals and detailed design analysis may also occur. Once a detailed vehicle design is complete, integration of subsystems with the vehicle is performed and system performance simulated. If the system performance meets specified performance levels, the design is accepted. If not, revisions throughout the design cycle may be introduced to develop a design which meets the required system performance.

This structure has a number of important strengths since it:

1. Provides a clear logical structure for the design and development process that moves from the abstract to the concrete;
2. Allows technical details to be added a level at a time in such a fashion that provides the encapsulation of detail needed to deal with complex systems;
3. Is a pipeline like process in which many different groups are productively active.

The difficulties of incorporating rapidly evolving computational approaches into this cycle are immediately apparent when one considers that:

1. The preliminary design process for an aircraft extends over 1 to 2 years;
2. Computational algorithms are usually developed for particular analysis problems not for integration into a multidisciplinary design process;
3. Design engineers are, properly, not specialists in numerical analysis or computer systems.

Under these conditions, design engineers are unable to effectively utilize the latest analysis techniques known to be effective in the research community. Therefore, the challenge is to create environments which facilitate knowledge transfer and provide a framework for creating and testing designs.

## Design Assistant Shell

In order to address this situation, a number of Knowledge Based System (KBS) shell implementations to aid design engineers are being considered. A KBS shell is a computing environment designed around similar applications and is a compromise between specific applications and general knowledge engineering tools.

One Design Assistant Shell architecture is illustrated in Figure 18. This architecture is a layered implementation which provides multiple user interfaces which serve the requirements of several different design team members. Figure 18 illustrates interfaces to: 1) CAD/CAM processors and their associated databases and specialized equipment, 2) network connections to local or remote computing resources such as CRAY class supercomputers or special purpose processors, 3) algorithm developers and their special knowledge, and 4) interactive control of design processing through the specification of design plans and stored procedural knowledge. The KBS shell provides assistance in generating and

controlling execution of the design plans, a common framework for expressing expert knowledge, automated execution of remote computational processing and display of computed results. The variety of user interfaces allows design team members with diverse backgrounds to communicate freely and isolates them from the complete design process complexity so that each may perform their tasks efficiently.

Information or knowledge about the design process is often procedural in nature and this fact is reflected in the choice of a design plan as a primary technique for knowledge representation and user interaction. A design plan is composed of:

- Objectives - what is to be achieved
- Constraints - checks and requirements to be satisfied
- Procedures - process to achieve objectives

Figure 19 illustrates a plan fragment that a design engineer might create to minimize wing drag. The most interesting section of this plan is the procedural knowledge expressed. This particular procedure creates an initial wing geometry through use of a library of known "good" pressure distributions and then passes control to a numerical optimizer to achieve an improved geometric design.

It is expected that the design engineer will be able to interactively create the procedures, call up and/or modify old procedures, or have procedures suggested to him.

In reference to Figure 19, it seems likely that a design engineer would have expert knowledge about "good" pressure distributions, but is unlikely to have expert knowledge about the Navier-Stokes algorithms used to evaluate drag or the numerical optimization procedure. In fact, the numerical optimization procedure of Figure 19 expands into the plan fragment illustrated in Figure 20. This optimization plan is a simple one based on linear combinations of approximating shape functions. A plan treatment of this type would be developed and maintained by algorithm specialists on the design team. Procedures in Figure 20 may also expand into plan fragments, but a successful plan expansion by the KBS plan generator will terminate in well-defined subproblems for which computational or analytical solutions exist. It may, of course, be impossible to successfully expand a design plan, or a procedure execution inside a plan may fail.

This section has reviewed techniques which can be used to simplify and manage the design process in a more efficient manner. These techniques provide a framework for creating and testing designs as well as mechanisms for the ultimate users to create and maintain their own knowledge bases. The multiple user interfaces mirror usual organizational structures but can relieve humans of tedious chores as they build and modify the knowledge base. In this way, knowledge transfer between design team members can be facilitated and more complete designs accomplished inside the available time and resources.

## SUMMARY

Computational physics has grown in influence in aircraft design over the last decade. Initial growth in this area was in the development and application of computational fluid dynamics, and today CFD remains the leading area for development and growth in computational physics. At the same time, the success of CFD has led to applications in other areas. Specifically, the similarities between the governing equations in CFD and CEM, as well as the importance of emerging design problems involving low observables, has led to major development efforts in the CEM area.

The utility and growth of computational methods in the design process depend on access to computational facilities with sufficient size and speed. Access to today's supercomputers, such as NAS, is essential in the exploration and development of future computational methods. The use of the resulting methods in a project design environment, such as at Northrop, requires exploration of alternative architectures, including parallel processing.

Finally, the depth of analysis and the volumes of data that will become available through the combination of more sophisticated methods and supercomputers will place unrealistic demands on the designer. The effective use of computational physics will therefore require the integration of expert systems. As computational engines continue to evolve, the value of fully-developed computational methods in aircraft design will increase proportionately. Within the next decade, we expect, computational methods will become the primary tools of the design engineer.

## REFERENCES

- Ballhaus, W. F.; Bailey, F. R.; and Frick, J.: Improved Computational Treatment of Transonic Flow About Swept Wings, *Advances in Engineering Sciences*, NASA CP-2001, 1976.
- Bayliss, A.; Goldstein, C. E.; and Turkel, E.: An Iterative Method for the Helmholtz Equation, *Journal of Computational Physics*, vol. 49, 1983, pp. 443-457.
- Bayliss, A.; Gunzburger, M.; and Turkel, E.: Boundary Conditions for the Numerical Solution of Elliptic Equations in Exterior Regions, *SIAM J. Appl. Math.*, vol. 42, no. 2, 1982, pp. 430-451.
- Bayliss, A.; and Turkel, E.: Far Field Boundary Conditions for Compressible Flows, *Journal of Computational Physics*, vol. 48, 1982, pp. 182-199.
- Bojarski, N. N.: K-Space Formulation of Electromagnetic Scattering Problem, *Air Force Avionics Laboratory Technical Report AFAL-TR-71-75*, March 1971.
- Bowman, J. J.; Senior, T. B. A.; and Uslenghi, P. L. E.: *Electromagnetic and Acoustic Scattering by Simple Shapes*, North-Holland Publishing Co., Amsterdam, 1969.
- Busch, Richard J.; Jager, Merle; and Bergman, Brad: The Application of Computational Fluid Dynamics to Aircraft Design, *AIAA Paper 86-2651*, October 1986.
- Denning, Peter J.: *The Science of Computing - 1985*, Research Institute for Advanced Computer Science, RIACS TR 85.12, September 1985.
- Erickson, G. E.: *Vortex/Linear Lift Augmentation*, AFWAL-TR- 85-3017, Volumes I-VI, June 1985.
- Erickson, Gary E.; Brandon, Jay M.: Low-Speed Experimental Study of the Vortex Flow Effects of a Fighter Forebody Having Unconventional Cross-Section, *AIAA Paper 85-1798-CP*, August 1985.
- Frenkel, Karen A.: Evaluating Two Massively Parallel Machines, *Comm. ACM*, vol. 29, no. 8, August 1986, pp. 752-758.
- Harrington, R. F.: *Field Computation by Moment Methods*, MacMillan, New York, 1968.
- Hwang, Kai: Multiprocessor Supercomputers for Scientific/Engineering Applications, *IEEE Computer*, vol. 18, no. 6, June 1985, pp. 57-73.
- Jameson, A.; Schmidt, W.; and Turkel, E.: *Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes*, *AIAA Paper 81-1259*, June 1981.
- Kaynak, Unver; Holst, Terry L.; Cantwell, Brian J.: *Computation of Transonic Separated Wing Flows Using an Euler/Navier-Stokes Zonal Approach*, NASA TM 88311, July 1986.
- King, R. W. P.; and Wu, T. T.: *The Scattering and Diffraction of Waves*, Harvard University Press, 1959.
- Ling, R. T.: *Numerical Solution for the Scattering of Sound Waves by a Circular Cylinder*, *AIAA Paper 86-1880*, July, 1986.
- Ling, R. T.: *Application of Computational Fluid Dynamics (CFD) Methods to Numerical Study of Electromagnetic Wave Scattering Phenomena*, *AIAA Paper 87-0487*, January, 1987.
- Mitra, R.: *Computer Techniques for Electromagnetics*, Pergamon Press, New York, 1974.
- Pulliam, Thomas H.: *Recent Improvements in Efficiency, Accuracy, and Convergence For Implicit Approximate Factorization Algorithms*, *AIAA Paper 85-0360*, January 1985.
- Rizzi, Arthur: Euler Solutions of Transonic Vortex Flows Around The Dillner Wing, *Journal of Aircraft*, vol. 22, no. 4, pp. 325-328, April 1985.
- Schneck, Paul B.; Austin, Donald; Squires, Stephen L.; Lehmann, John; Mizell, David; and Wallgren, Kenneth: *Parallel Processor Programs in the Federal Government*, *IEEE Computer*, vol. 18, no. 6, June 1985, pp. 43-56.
- Umashankar, K.; and Taflove, A.: A Novel Method to Analyze Electromagnetic Scattering of Complex Objects, *IEEE Transactions on Electromagnetic Compatibility*, EMC-24, 1982, pp. 397-405.

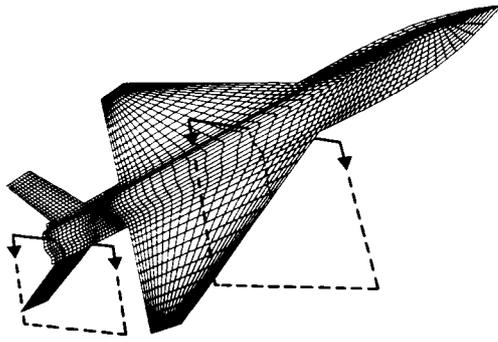


Figure 1a. Surface grid for wing-body-tail configuration

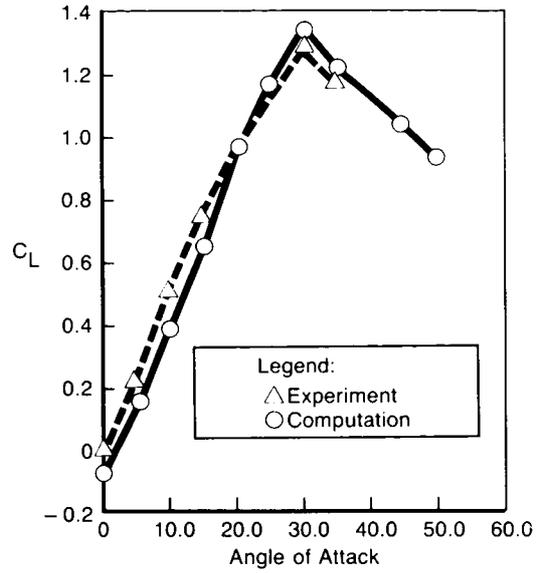


Figure 2. Comparison of test results and Euler calculations for the AFVFM

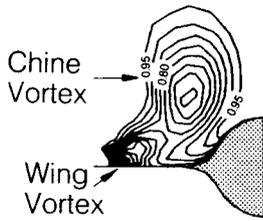


Figure 1b. Total pressure contours at wing-body station

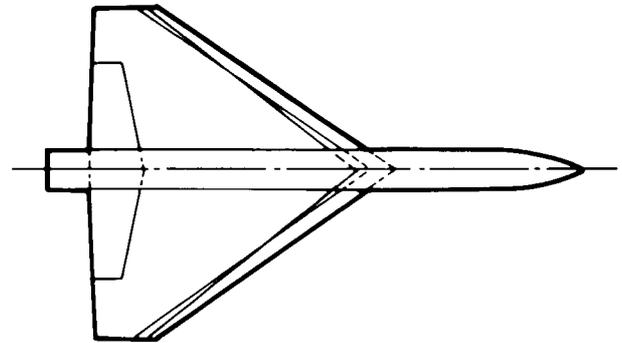


Figure 3. Planform view of AFVFM

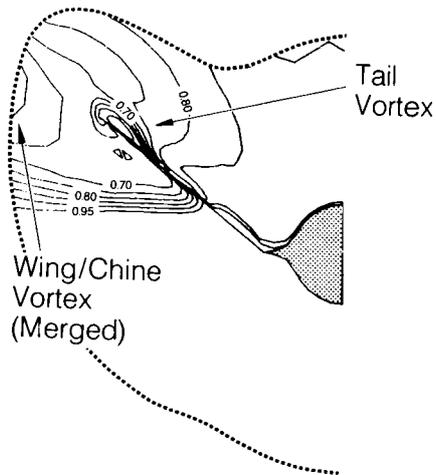


Figure 1c. Total pressure contours at tail-body station

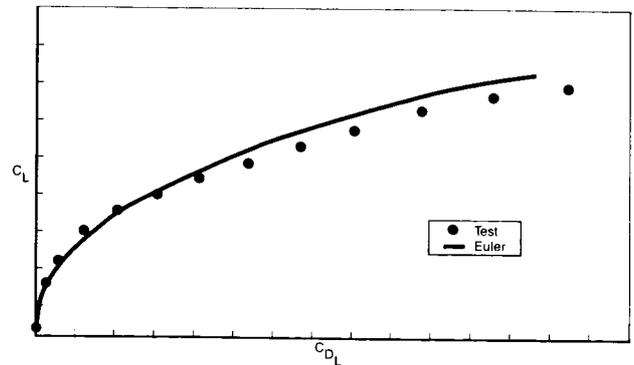


Figure 4. Comparison of F-20 Euler results with test force and moment data at Mach 0.80

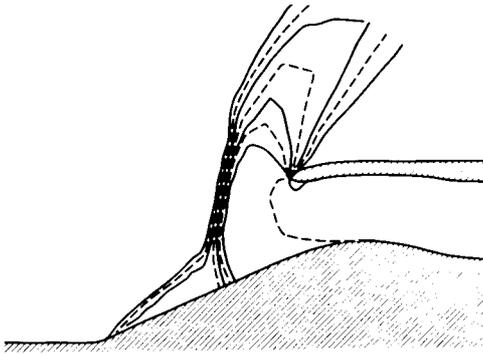


Figure 5. Pressure contours for a compression ramp inlet at Mach 2.0

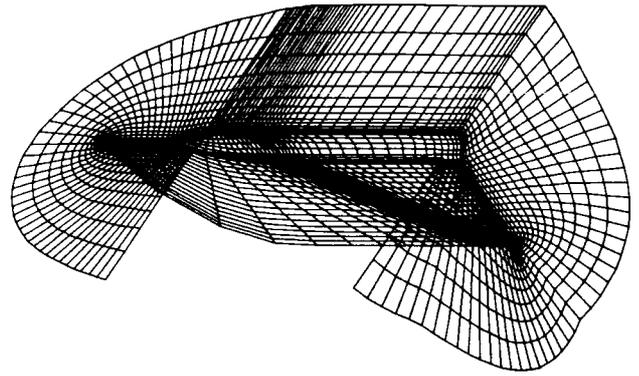


Figure 8. Grid for AFVFM (figure 3)

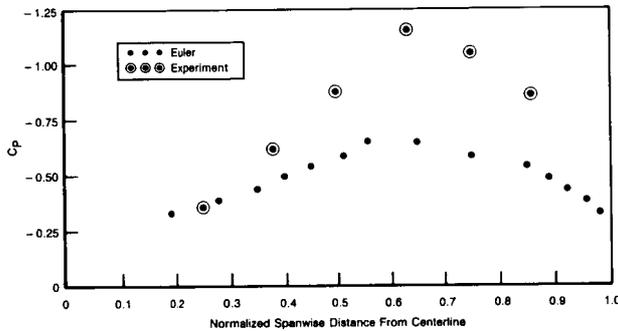


Figure 6. Comparison of test results and Euler calculations at a forebody station for the wing-body-tail configuration shown in figure 1a

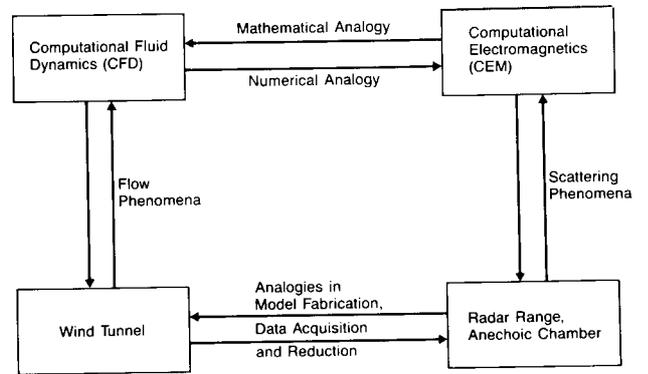


Figure 9. Aerodynamics and Electromagnetics in the design process

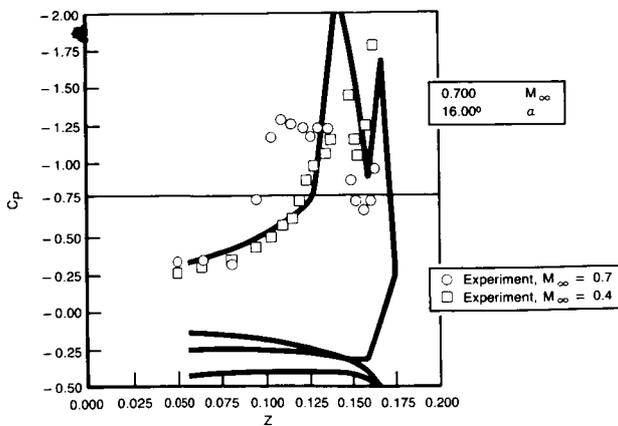


Figure 7. Comparison of test results and Euler calculations at a wing body station for the AFVFM (figure 3)

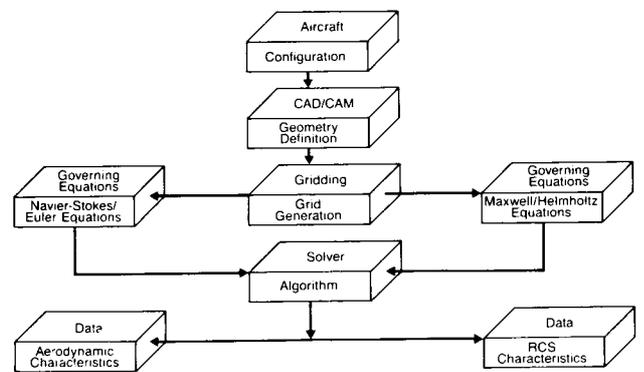


Figure 10. Unified solution of aerodynamic and RCS problems

Boundary Value Problems	Computational Fluid Dynamics (CFD)	Computational Electromagnetics (CEM)
Governing Equations	Navier-Stokes/Euler Equations	Maxwell/Helmholtz Equations
Outer Boundary Condition	Freestream Condition	Radiation Condition
Inner Boundary Condition	Flow Tangency Condition	Media Interface Condition

Figure 11. Analogy between Computational Fluid Dynamics (CFD) and Computational Electromagnetics (CEM)

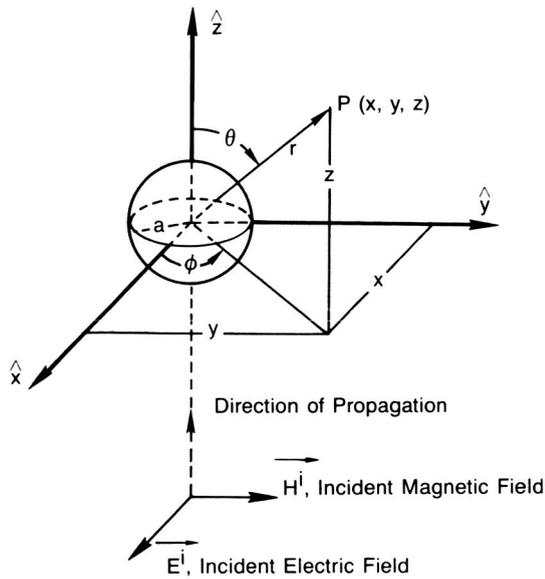


Figure 12. Plane wave incidence on a sphere and the coordinate systems

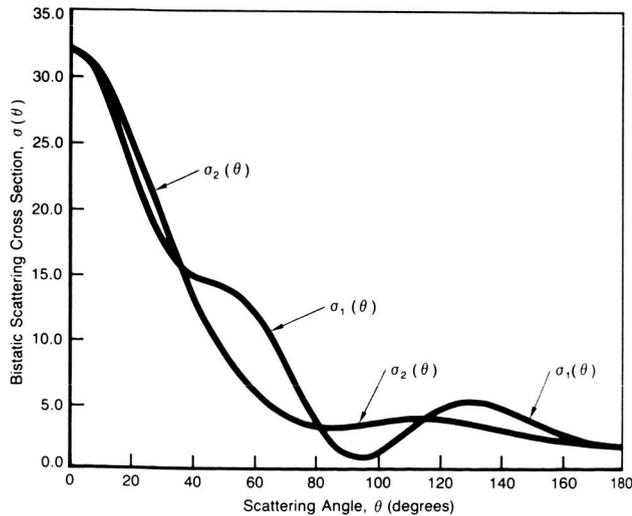


Figure 13. Distribution of bistatic scattering cross sections for a perfectly conducting sphere in a plane wave field at  $ka = 2.9$ ;  $\sigma_1(\theta)$  is in the E-plane;  $\sigma_2(\theta)$  is in the H-plane

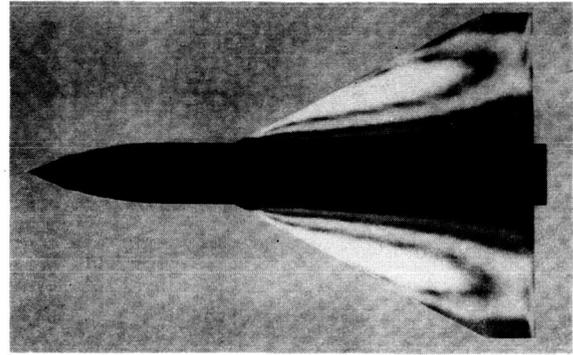


Figure 14. Cross-flow velocity magnitudes on a 3-D fighter configuration

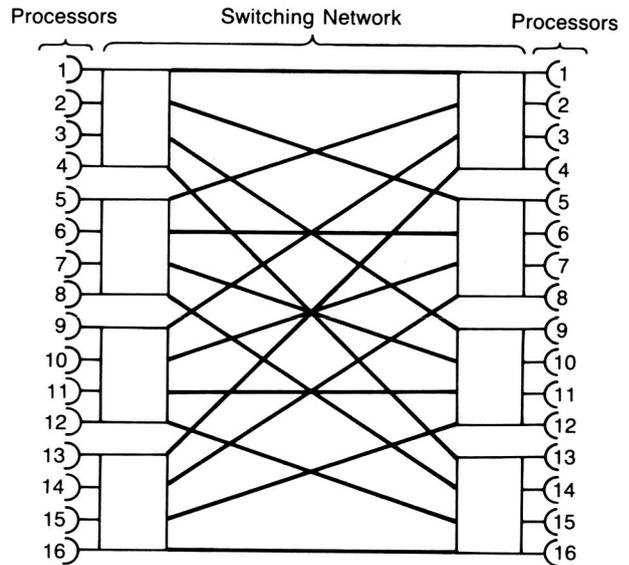


Figure 15. 16 processor system with butterfly switching network interconnect

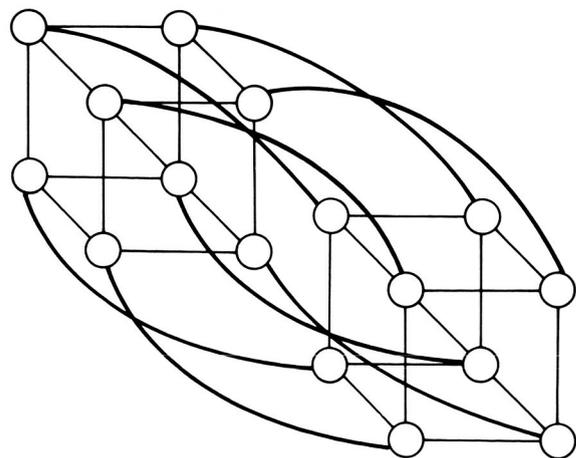


Figure 16. 16 processor system with 4-dimensional hypercube interconnect

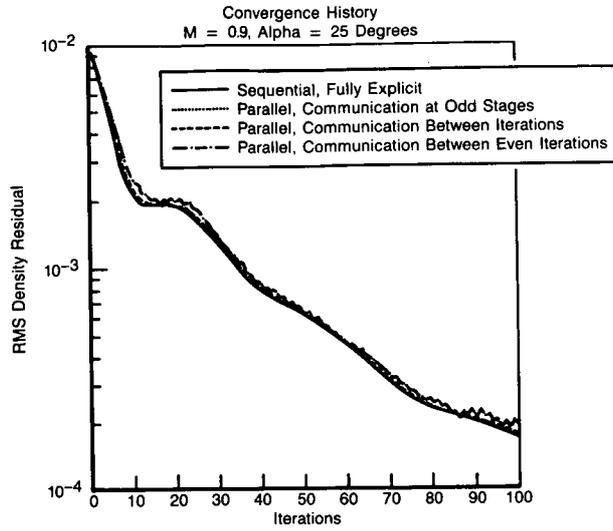


Figure 17. Finite-volume Euler algorithm convergence with varying levels of interprocessor communications

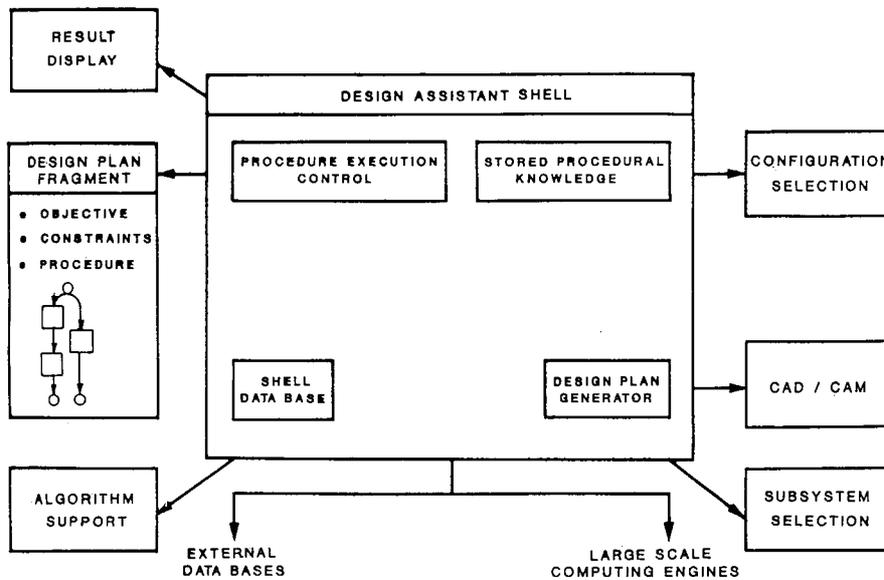


Figure 18. Design assistant shell logical architecture. Multiple interfaces facilitate design plan interaction.

OBJECTIVE: MINIMIZE DRAG AT  $M = 0.9$

CONSTRAINTS:  $C_L = 1.0$

PROCEDURES:

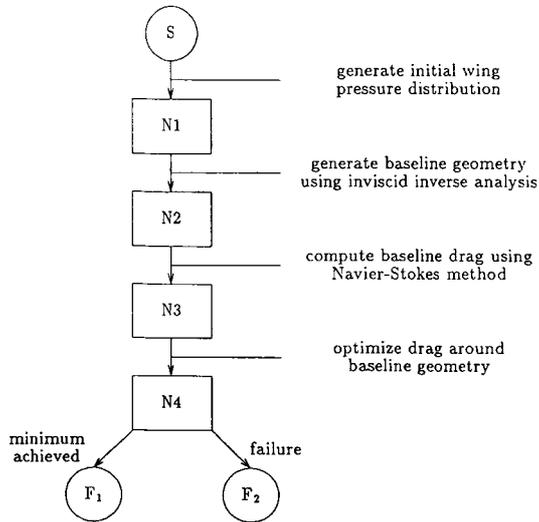


Figure 19. Design plan fragment for drag minimization step

OBJECTIVE: MINIMIZE PERFORMANCE MEASURE

CONSTRAINTS: SCALAR PERFORMANCE MEASURE, BASELINE GEOMETRIC SPECIFICATION, TARGET PARAMETERS  $\langle C_L \dots \rangle$

PROCEDURES:

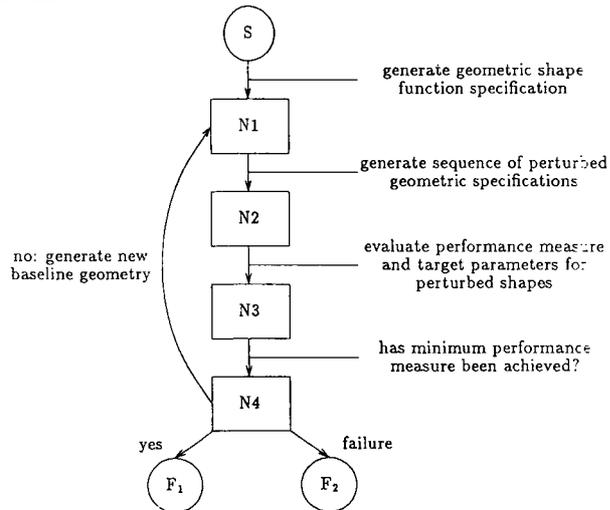


Figure 20. Design plan fragment for generalized optimization procedure